Intelligence Community and Department of Defense Content Discovery & Retrieval Integrated Project Team (CDR IPT)

IC/DoD SOAP Interface Encoding Specification for CDR Deliver v1.0

12 May 2011

REVISION/HISTORY

Doc Revision	Revised By	Revision Date	Revisions
0.1	CDR IPT	13 September 2010	Initial draft for subgroup review.
0.2	McIntyre, Wigglesworth	12 October 2010	Revised following subgroup review, and again following the 5 October CDR-IPT Design Meeting.
0.3	CDR IPT	18 October 2010	Revised following subgroup review
1.0-20101020	CDR IPT	20 October 2010	Revised following subgroup review
1.0-20101026	Wigglesworth	26 October 2010	Tech Edit Recommendations
1.0-20101027	Rothrock	29 October 2010	Tech Edit Recommendations
1.0-20101223	Coston/Wigglesworth	12 January 2011	Reconciliation with REST Deliver Spec
1.0-20110318	Coston/Wigglesworth	18 January 2011	Responses to community review
1.0-20110429	CDR IPT	29 April 2011	Responses to community review

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

1.1 Service Overview

This document defines requirements and provides guidelines for the realization of the Content Discovery and Retrieval (CDR) Deliver Component as a web service using the Simple Object Access Protocol (SOAP) style binding, hereafter termed a ***Deliver Service*** in this document. It describes a ***Deliver Service's*** interface and other aspects in detail, providing enough information for ***Deliver Service*** providers and implementers to create CDR-compliant Deliver ***Services***.

The ***Deliver Service***, as defined by the Intelligence Community/Department of Defense (IC/DoD) CDR Specification Framework, serves as a "push" mechanism to send information resources. The ***Deliver Service*** relies on mechanisms that are already well established in the service oriented architecture design and development community. Specific mechanisms include, but are not limited to:

- Simple Object Access Protocol (SOAP) [1]
- Web Services Definition Language (WSDL) [2]
- Web Services Addressing [4][5]

The ***Deliver Service,*** as defined, supports the delivery of a specified resource payload directly to a consumer specified location. In its simplest form, ***Deliver*** will take a consumer-supplied payload and send it to another consumer as specified by the delivery destination and properties. The ***Deliver Service*** may include additional (interim) processing, including but not limited to compression, encryption, or conversion. The specification of interim processing is beyond the scope of this document.

The delivery destination can be:

- A specified location (e.g., file transfer protocol (ftp) folder, shared drive)
- A receiving component implementation
- Another component or service endpoint within the architecture

The implementation method is left to the implementers of the ***Deliver Service.*** The Deliver Specification focuses on a single delivery target, but it does not preclude an implementation having multiple delivery targets. The consumer requesting the delivery may want to obtain the status of the Deliver Function, especially in scenarios where the delivery content is not returned directly to the requestor; in the initial version of this Deliver Specification, we demonstrate status as an output returned to the requestor. However, future versions of this specification may provide other methods for obtaining status.

1.2 Relationship to Other CDR Architecture Elements

The CDR Architecture prescribes an abstract-to-concrete model for the development of architecture elements and guidance for CDR. Each layer, or tier, of the model is intended to provide key aspects of the overall guidance to achieve the goals and objectives for joint DoD/IC content discovery and retrieval. The following graphic in Figure 1, discussed in detail within the CDR Reference Architecture (RA), illustrates this model.

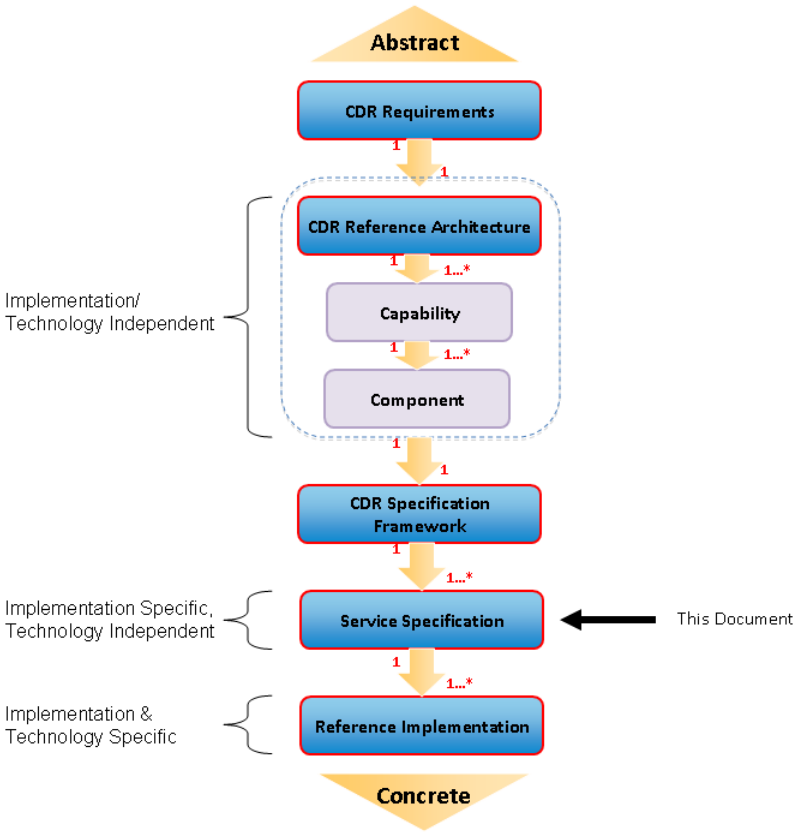

Figure 1 - CDR Architectural Model

As illustrated in Figure 1, the Specification Framework derives from the RA and describes behavior in terms of the capabilities, components, and usage patterns defined in the RA. The Specification Framework expands on the details of information flows and the information conveyed in those flows to provide a consistent basis for multiple Service Specifications to provide consistent interfaces, both in terms of the structure and of the semantics of the exchanged information. Service Specifications, such as this one, provide implementation-specific guidance. More specifically, this Deliver Specification defines the specific guidance for implementing the Deliver ***Service***

1.3 Scope

As shown in the shaded area in Figure 2, below, this specification is limited to the description of the interaction between the Initiating Consumer and the ***Deliver Service***. The association between these two components is depicted in the diagram as a ***Deliver***

Service invocation with a set of parameters that includes (payload, properties, and destination). Components, Interactions and associations that lie outside the shaded areas clarify the overall design and provide a context for the use of deliver. Interactions/associations that are outside the shaded rectangle are used in this document to clarify the interaction between the Initiating Consumer and the ***Deliver Service***.

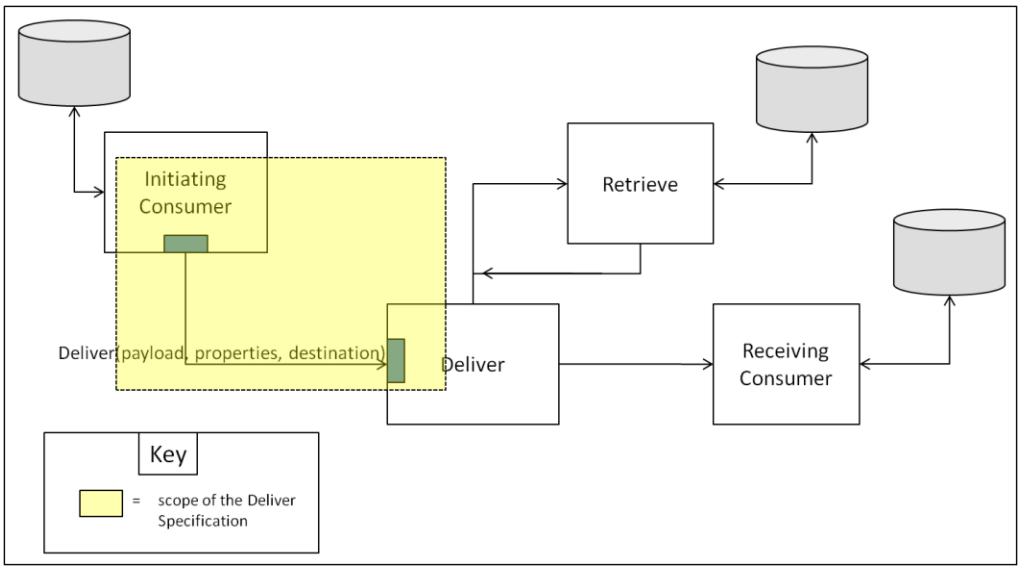

Figure 2 - Scope of the Deliver Service Specification for SOAP Implementations

This specification covers the following aspects of a SOAP-based ***Deliver Service***:

- **Service Interface** defines the base SOAP constructs to expressing inputs, outputs, and faults
- **Implementation** provides additional implementation guidance beyond the behavior and interface guidance
- **Reference Documentation** provides references to other CDR and community artifacts (e.g., Service Security Reference Architecture [3])

1.4 Notational Convention

The key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this specification are to be interpreted as described in the IETF RFC 2119. When these words are not capitalized, they are meant in their natural-language sense.

When describing concrete XML schemas and exemplary XML documents, this specification uses XPath as the notational convention. Each member of an XML schema is described using an XPath notation (e.g., /x:RootElement/x:ChildElement/@Attribute). The use of {any} indicates the presence of an element wildcard (<xs:any/>). The use of @{any} indicates the presence of an attribute wildcard (<xs:anyAttribute/>).

Examples in this text are distinguished by a black border. These are meant to be illustrative and represent one way that the described syntax can be used.

1.5 Conformance

This specification defines an interface to a ***Deliver Service*** to which an implementation MUST conform. For an implementation to conform to this Deliver specification, it MUST adhere to all mandatory aspects of the specification.

1.6 Namespaces

The following table represents only those XML namespaces that are directly leveraged in this document.

Table 1 – Referenced XML Namespaces

Prefix	URI	Description
soap	http://www.w3.org/2003/05/soap-envelope	W3C SOAP Version 1.2
wsa	http://www.w3.org/2005/08/addressing	WS-Addressing Definition
wsaw	http://www.w3.org/2006/05/addressing/wsdl	WS-Addressing – WSDL Binding
cdrd	urn:cdr:1.0:soap:deliver	CDR v 0.1 Deliver Specification for SOAP Implementations

2 Deliver Service Behavior

This section uses basic message exchange patterns to clarify the behavior of the Deliver Service Component in the context of the CDR architecture components.

2.1 Component Interactions

The ***Deliver Service*** supports a fundamental message exchange pattern. The pattern, shown in Figure 3, reflects a case where the Initiating Consumer supplies the information content via the payload parameter to be delivered to the Receiving Consumer, as specified via the destination parameter. The properties included as part of the ***Deliver Service*** request may be used to provide delivery specific information, including but not limited to interim processing, routing, and security.

In the event that a particular implementation of the ***Deliver Service*** makes use of "default" values for message retrieval and/or delivery, the service implementer is responsible for publishing this information using an agreed upon mechanism. Cases where information is not supplied as part of the Deliver Service Request and service defaults are not available will result in a fault condition.

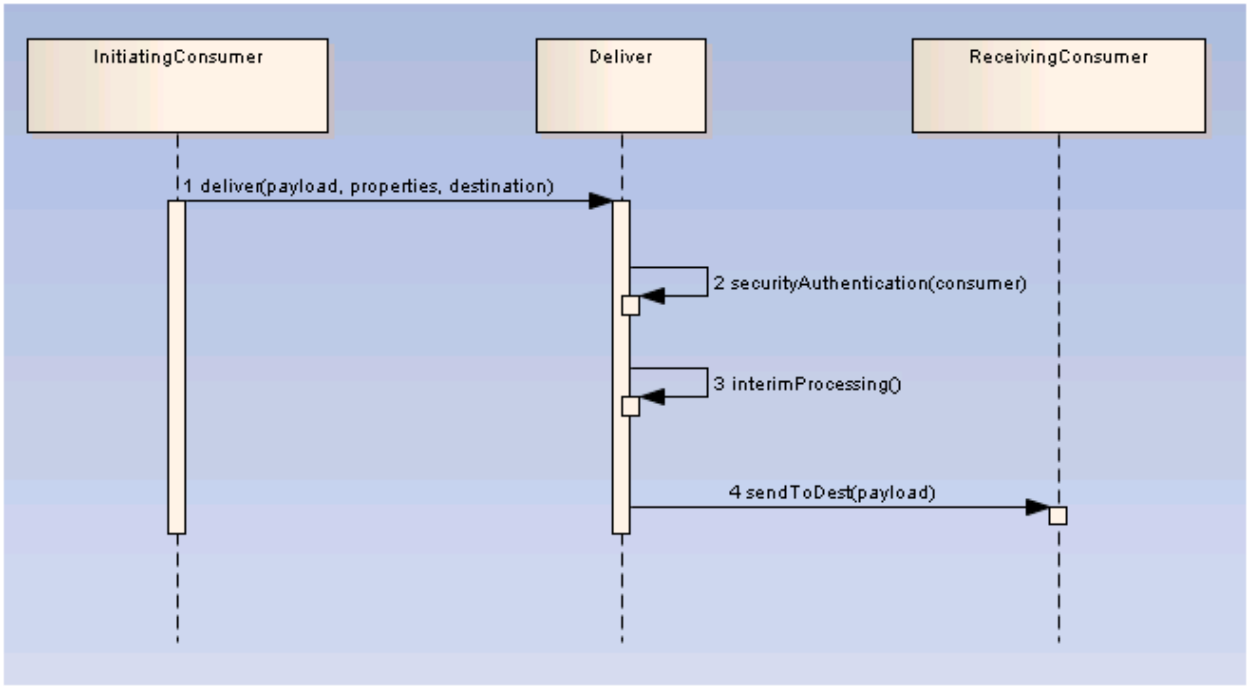

Figure 3 - Deliver: Payload Provided by the Initiator

Step 1 – Initiating Consumer sends a Deliver Service Request to the ***Deliver Service***.

Step 2 – The ***Deliver Service*** leverages a set of security components to verify that the Initiating Consumer is authenticated and authorized to send a specific information resource (payload) to the Receiving Consumer; and the Receiving Consumer is authenticated and authorized to receive the specified information payload. The Joint

IC/DoD Security Reference Architecture [3] defines the specific security components and interactions needed to perform this verification.

Step 3 – The optional interim processing may represent internal capabilities of the Deliver implementation or may be external capabilities for which the Deliver implementation acts as a consumer. The ***Deliver Service*** implementation is NOT required to include any interim processing (e.g., applying compression algorithms or translating the payload to a different format).

Step 4 – Payload is delivered to Receiving Consumer.

2.2 SOAP Specific Behavior Information

This Deliver Specification allows for the flexibility to use other protocols and options (SOAP over Java Messaging Service (JMS), guaranteed delivery, etc.) and still be in conformance with this specification even if they are not used in the examples provided.
The ***Deliver Service***, as defined in the specification framework, supports the input of Retrieve properties (*instead* of a resource payload). This behavior is not necessary in a SOAP implementation, as it is covered as part of the WS-Addressing specifications [4][5].

2.3 Functional Behavior

The ***Deliver Service*** is REQUIRED to function as described by the CDR Specification Framework with any input, behavior, output, and fault condition extensions listed below.

Function	Input	Output	Fault
Deliver	*cdrd:DeliverTo, {Deliver Properties}, {Resource Payload}*		Defined within CDR Framework

3 Deliver Service Interface

3.1 Input

The following table shows each input variable defined in the ***Deliver Service's*** Deliver function, and maps each to the ***Deliver Service*** variables as defined in the IC/DoD CDR Specification Framework (see Section 5.1 of this document).

Table 2 - Deliver Specification Input Variables

Input Variable	Required/Optional
wsa:Action	Required
cdrd:DeliverTo	Optional[1]
{Deliver Properties}	Optional
{Resource Payload}	Required

The following example illustrates the mechanism for providing input in the Deliver request message. In the example, the *wsa:Action* is inserted in the SOAP Header along with the *cdrd:DeliverTo* element and any *{Deliver Properties}*. The *{Resource Payload}* is inserted into the SOAP Body.

Example 1: Deliver Request Message with {Resource Payload} in the SOAP Body

```
<soap:Envelope>
  <soap:Header>
    ...
    <wsa:Action>urn:cdr:1.0:soap:action:deliver</wsa:Action>
    <cdrd:DeliverTo>
      <wsa:Address>http://ExampleDestinationAddress</wsa:Address>
      <wsa:Metadata >
        <wsaw:InterfaceName>ExampleDestinationPortType</wsaw:InterfaceName>
        <wsaw:ServiceName EndpointName=”ExampleDestinationPortName”>
          ExampleDestinationService
        </wsaw:ServiceName>
      </wsa:Metadata>
    </cdrd:DeliverTo>
   {Deliver Properties}
    ...
  </soap:Header>
  <soap:Body>
    {Resource Payload}
  </soap:Body>
</soap:Envelope>
```

The following is a description of significant elements:

3.1.1 The Action Element (soap:Envelope/soap:Header/wsa:Action)

Required. This element indicates the intent of the message. In the ***Deliver Service's*** Deliver function, the value of this element MUST always be "urn:cdr:1.0:soap:action:deliver."

[1] *If a DeliverTo Element is not specified, the component may deliver the {Resource Payload} to a default recipient*

3.1.2 The DeliverTo Element (soap:Envelope/soap:Header/cdrd:DeliverTo)

Optional[2]. This element provides a recipient to which the {Resource Payload} should be delivered. The DeliverTo Element is a wsa:EndpointReferenceType and MUST contain the endpoint information necessary for referencing a specific service endpoint. Additional information on specifying endpoint information can be found in the Web Services Addressing 1.0 – Core specification [4].

If the DeliverTo recipient is not recognized by the ***Deliver Service***, it must return an Unknown Recipient Fault.

3.1.3 Deliver Properties (soap:Envelope/soap:Header/{Deliver Properties})

Optional. This is a placeholder for any properties or special handling instructions required for the specific implementation for this ***Deliver Service*** to distribute the {Resource Payload} to the Recipient.

3.1.4 Resource Payload (soap:Envelope/soap:Body/{Resource Payload})

Required. This is a placeholder for the Resource Payload being delivered to the Recipient by the ***Deliver Service*** implementation.

The Resource Payload MAY be the content resource or metadata describing the content resource. The Resource Payload MAY be inserted directly in the SOAP:Body or in an XML Wrapper such as an Atom:Feed, Atom:Entry, or WS-Notification payload.

If the {Resource Payload} contains metadata describing a content resource, it SHOULD contain a reference to that content resource to facilitate retrieval.

3.2 Output

Although there are no required output variables in the ***Deliver Service's*** Deliver function, a specific ***Deliver Service*** implementation MAY choose to return delivery-specific information (such as a reference to the deliver status) to the ***Deliver Service*** initiator.

3.3 Fault Conditions

An implementation of the ***Deliver Service*** MUST allow for the Fault Conditions defined in the CDR Specification Framework.

[2] *If a DeliverTo Element is not specified, the component may deliver the {Resource Payload} to a default recipient*

3.3.1 Fault Handling in SOAP

Different versions of SOAP may have different fault handling syntaxes. ***Deliver Services*** MUST use the primary fault handling mechanism for the version of SOAP they support and to which the service is bound. In the following example, an Unsupported Identifier Type fault is returned using the SOAP 1.2 syntax:

```
<soap:Fault>
  <soap:Code>
    <soap:Value>soap:Sender</soap:Value>
  </soap:Code>
  <soap:Reason>
    <soap:Text>Identifier Execution Fault</soap:Text>
    <soap:Text>Service could not retrieve the specified resource</soap:Text>
  </soap:Reason>
</soap:Fault>
```

3.4 SOAP Version

A ***Deliver Service*** MUST use SOAP 1.2.

3.5 WSDL

The Web Service Description Language (WSDL) document that specifies the bindings for the Deliver Service is implementation specific. A WSDL template that provides an initial reference for WSDL development is provided as a supplement to this specification.

4 External Dependencies

4.1 Service Security

The Security focus area provides a set of security-focused services to the IC and DoD for protecting access to services, data, and their interactions within the IC/DoD Enterprise. Integration of Security capabilities is advocated, both from the service discovery and the service access standpoint, to protect content providers and consumers from attack from any unknown entities. Security capabilities are responsible for authenticating and authorizing of consumers and consumer agents, binding IA metadata to the information that it describes (query, search result, or retrieved content), controlling access to content resources, and enabling cross-domain search and retrieval. Furthermore, security capabilities provide integrity, confidentiality, and audit services that CDR providers can leverage. CDR providers together with their security engineers should reference the Joint IC/DoD Security Reference Architecture (SRA) [3] for guidance on integrating and using the security services within and between CDR components[3]. It is expected that the SRA and derived specifications will provide guidance for implementers of the CDR components which identifies interface points for requesting security services. As appropriate, this guidance will be documented within the CDR Architecture Model to achieve secure CDR services.

4.1.1 Service Security Concerns

The following security relevant considerations are consolidated in this section to more clearly define points of intersection and dependency upon the Joint IC/DOD Security Reference Architecture (SRA) [3] that may be of importance in realizing the CDR Compliant Services:

- Identification and Authentication: The operations defined here require the Consumer Component to provide an authenticated identity to the CDR Component it is calling. The authentication requirement extends to authenticating CDR Components acting on behalf of a consumer (chained authentication).
- Activity Authorization: The CDR Component must determine if the authenticated consumer is authorized to perform the requested activity. In addition, it must determine if the intended recipients of delivered metadata or resource content are authorized to receive it.
- Access Control: The CDR Component must abide by the access control policies for search results and retrieved content based on their IA Metadata, and on Consumer and CDR Component security attributes. Access control is applied to both the Content Collection and individual Content Resources within the Collection.
- Classification: General rules and specifications referring to the classification of saved resources also apply to CDR Components, but are not described in this framework.

[3] *This guidance could also cover the security aspects of CDR components interacting with non-CDR components.*

- Auditing and Logging: General rules and specifications referring to the auditing and logging of data apply to CDR Components, but are not described in this framework.
- Protecting confidentiality, integrity, availability and non-repudiation: General rules and specifications referring to these security concerns apply to CDR components, but are not described in this specification. This includes message level and transport level security.

4.1.2 Security Fault Conditions

The following potential security fault conditions are common to most of the CDR capabilities:

- Action Not Authorized: The Consumer does not have permission to perform the requested function on the requested resource.
- Identity Not Authenticated: The Consumer could not be authenticated. (The claimed identity could not be confirmed.)

5 References

This section includes additional references that may be used to provide further insight into the overall design concepts that serve to guide the CDR-IPT engineering efforts.

5.1 Content Discovery and Retrieval Specifications

The CDR Reference Architecture and Specification Framework provide essential background and context to service designers. This document was based on the following CDR Reference Architecture and Specification Framework document versions:

- *"IC/DoD Content Discovery and Retrieval Reference Architecture Version 1.0",* 19 December 2009
- *"IC/DoD Content Discovery and Retrieval Specification Framework Version 0.9"* 6 June 2010

The most recent version of the documents can be found at the unclassified Intelink web site [6].

5.2 Additional References

[1] Simple Object Access Protocol (SOAP) - http://www.w3.org/TR/soap12-part1/

[2] Web Services Definition Language (WSDL) - http://www.w3.org/TR/wsdl

[3] Intelligence Community and Department of Defense Service Security Working Group, Joint IC/DoD Security Reference Architecture, available via Intelink.

[4] World Wide Web Consortium, *Web Services Addressing 1.0 - Core*, 9 May 2006.

[5] World Wide Web Consortium , *Web Services Addressing 1.0 - WSDL Binding*, 16 February 2006.

[6] Unclassified Intelink web site – https://www.intelink.gov/site/odni/cio/i2e/focus/iads/cdript/default.aspx